P515881.PDF [Page: 3 of 37]

CR 2001-028

Analysis of SIMD Signal Processing Performance

Authors:
Bill Brinson (VPI)
Ron Saper (VPI)

February 2001

DEFENCE R&D DÉFENSE
Defence Research Establishment Ottawa

Vantage Point INTERNATIONAL

Vantage Point International Inc
1704 Carling Avenue 2nd floor
Ottawa, Ontario, K2A 1C7
Tel: 613-798-2244 ext. 224, 221 Fax: 613-798-5369
www.vantpoint.com

©HER MAJESTY THE QUEEN IN RIGHT OF CANADA
AS REPRESENTED BY THE MINISTER OF NATIONAL DEFENCE, 2001

102-01791/

REPRODUCTION QUALITY NOTICE

This document is the best quality available. The copy furnished to DRDCIM contained pages that may have the following quality problems:

: Pages smaller or Larger than normal

: Pages with background colour or light coloured printing

: Pages with small type or poor printing; and or

: Pages with continuous tone material or colour photographs

Due to various output media available these conditions may or may not cause poor legibility in the hardcopy output you receive.

☒ If this block is checked, the copy furnished to DRDCIM contained pages with colour printing, that when reproduced in Black and White, may change detail of the original copy.

Analysis of SIMD Signal Processing Performance

Authors:
Bill Brinson (VPI)
Ron Saper (VPI)

February 2001

Vantage Point International Inc
1704 Carling Avenue 2nd floor
Ottawa, Ontario, K2A 1C7
Tel: 613-798-2244 ext. 224, 221 Fax: 613-798-5369
www.vantpoint.com

Contract# W7714-0-0350

Prepared for : Mr. Robert Inkol
Defence Research Establishment Ottawa
National Defence
Ottawa, Ontario, K1A 0Z4

©HER MAJESTY THE QUEEN IN RIGHT OF CANADA
AS REPRESENTED BY THE MINISTER OF NATIONAL DEFENCE, 2001

P515881.PDF [Page: 6 of 37]

ABSTRACT

The Intel Pentium 3 processor offers the potential for improved performance for floating point intensive algorithms which benefit from a high degree of parallelism. This is typical of signal processing applications. High performance may be achieved using Intel's Streaming SIMD Extensions (SSE). The SIMD (Single Instruction Multiple Data) instructions utilize the additional eight 128-bit XMM registers on the P3 processor, supplying the software developer with specialized SIMD instructions which accomplish multiple floating point arithmetic operations simultaneously at high throughput.

SSE instructions offer advantages to the software developer, however, there are a number of software factors that affect performance, notably data alignment and data dependancy. Intel offers an excellent signal processing library which allows the user to take advantage of SSE without specific SSE programming knowledge.

Although increased performance is possible through use of SSE, unless a number of hardware factors are taken into account the potential advantages offered by SSE will be nullified. Factors such as data transfer rates, data dependancies, processor clock /bus speeds, cache size and cache speed have a significant impact on signal processing performance.

This report deals with signal processing algorithms relevant to electronic support measures and issues affecting their performance. These algorithms were implemented using SSE and timed extensively on a variety of machines for various data lengths. We used two FFT implementations and a benchmark sequence of operations for signal detection processing. The results are analysed to determine which operations take the most time and which factors limit performance. Finally, conclusions and recommendations are provided.

P515881.PDF [Page: 7 of 37]

EXECUTIVE SUMMARY

Electronic Support Measures (ESM) research at the Defence Research Establishment Ottawa (DREO) relies heavily on computer analysis of digital signals. While general purpose desktop microcomputers are generally used to develop algorithms, until recently this class of computer has lacked the power to efficiently perform digital signal processing at or near real-time rates. Recent developments, however, have made computing power less costly and more accessible, although some extra effort is needed to harness the available processing power of the latest generation of desktop computers for scientific software development. The work described in this report aims to investigate the performance available using the SIMD features of the Pentium 3 processor for possible application to DREO's MIlitary Data Analysis System (MiDAS).

Intel's Streaming SIMD Extensions (SSE) can be used to improve signal processing performance if the application has been optimized for its use. Intel offers a signal processing library (SPL) which the software developer can use to perform this optimization by simply calling pre-written functions for standard operations. The Intel compiler's SIMD intrinsic functions offer the developer an alternate way to take advantage of the P3's eight additional 128-bit XMM registers to perform parallel operations with fewer instructions.

For the 1K and 2K point FFTs of greatest interest, the real, in-place FFT from Intel's Signal Processing Library was well suited in terms of both performance and ease of implementation. As an alternative, a complex, in-place FFT was developed using a Split Radix algorithm derived from the Intel Architecture Performance Training Library (v. 4.5). The SRFFT (Split Radix FFT) was slower than the Intel SPL complex, in-place FFT for N = 2K or smaller FFTs. However, at N > 4K the SRFFT displayed superior performance.

The 512 and 256 KB L2 caches of the P3 processors were not limiting factors for the 2K or smaller FFTs. The 32 KB L1 cache was not exceeded until a N = 4K FFT was performed. Once L2 cache is exceeded, at N = 64K and N = 32K FFTs for the 512 KB and 256 KB L2 cache processors respectively, data transfer becomes a limiting factor on performance due to cache misses.

Data transfer is also a factor when dynamic data sets are used. This is due to the fact that the data must first be loaded into cache from memory or an input source. Results have shown that once data has been cached, main memory data transfer is eliminated or drastically reduced. Besides the FFT speed tests, a representative application was benchmarked. The benchmark application performs three operations on data. These are windowing (hamming), real FFT and a cross spectrum. The application suffers a delay due to memory data transfer at the windowing operation. This accounts for approximately 30% of execution time for the data length of interest. There is no additional main memory data transfer overhead when performing the FFT or cross spectrum operations after the window is applied due to caching. Since this type of data transfer is within the scope of interest, an improvement can be obtained by acquiring a processor with a faster front-side bus speed and/or by implementing prefetching.

P515881.PDF [Page: 8 of 37]

For the MiDAS application using current technology, we recommend that a fast P3 processor with 256 KB L2 cache and a 133 MHz bus be used. In the future, increasing processor clock rates and fast bus speeds will allow increases in signal processing speed. The higher clock rates will be supported for only the more recent version of the P3 which has the 256 KB L2 cache. The Intel SPL offers impressive performance for the critical 2K FFTs and below, and it is easy to implement. If the achieved performance is found to be inadequate or marginal, the Athlon processor from AMD should be investigated before major software investments are made in the P3. Improved performance may be achieved through prefetching of data to reduce initial data transfer and by further software development. The software developments could include attempts to implement faster FFTs specialized for the application of interest.

P515881.PDF [Page: 11 of 37]

APPENDIX A: Test Summary Table 21

APPENDIX B: Platform Specifications 23

LIST OF TABLES

Table 1: Pentium 3 Processor Cache and Bus Speed Types 2
Table 2: Summary of Benchmark Times for Two P3 Systems 3
Table 3: Intel SPL Functions of Interest 4
Table 4: Intel SPL Power and Phase Functions 19

LIST OF FIGURES

Figure 1: Benchmark Timing Types 10
Figure 2: Comparison of Expanded Split Radix and Intel Complex FFTs 11
Figure 3: Intel Real FFT Performance for Four Processors, L2 Cache Disabled 13
Figure 4: Intel Real FFT Performance for Four Processors, L2 Cache Enabled 14
Figure 5: Performance of Components of the Benchmark Processing Sequence 15
Figure 6: Relative Importance of Components of the Benchmark Sequence 16

P515881.PDF [Page: 10 of 37]

TABLE OF CONTENTS

ABSTRACT iii

EXECUTIVE SUMMARY iv

LIST OF TABLES vii

LIST OF FIGURES vii

1 Introduction 1
- 1.1 Scope 1
- 1.2 Background 1
- 1.3 Summary of Benchmark Timings 2

2 Windowing , FFT Types and Cross Spectrum 3
- 2.1 Overview 3
- 2.2 Windowing 4
- 2.3 SPL FFTs 4
- 2.4 Extended Split Radix FFT 5

3 Software Effects on Speed 5
- 3.1 Cache and Pre-Fetching 5
- 3.2 Data Alignment in Memory 5
- 3.3 Streaming Stores 6
- 3.4 Order of Operations 6

4 Hardware Effects on Speed 6
- 4.1 Cache Size 6
- 4.2 Processor Clock Speed 7
- 4.3 Processor Bus Speed 7
- 4.4 Data Transfer Issues 7

5 Testing and Results 8
- 5.1 Overview 8
- 5.2 Expanded SRFFT vs Intel SPL Complex FFT 11
- 5.3 Cache and Clock Speed Effects for Intel Real FFT 12
- 5.4 Performance of Typical MiDAS Processing Sequence 15

6 Conclusions 17

7 Recommendations 18

ACRONYMS, ABBREVIATIONS, DEFINITIONS 19

REFERENCES 19

1 Introduction

1.1 Scope

This report investigates the application of SIMD features of the Pentium 3 processor to the design and optimization of signal processing algorithms. We discuss both software and hardware issues pertaining to computer systems based on the P3 processor.

The analysis in this report is based on prior knowledge, some general assumptions and the results of numerous tests. The software focus has been placed on FFTs, windowing (hamming) and cross spectrum operations. The focus of hardware testing has been placed on cache size, clock speed, bus speed and data dependency/transfer issues.

1.2 Background

DREO requires investigation into the design and optimization of signal processing algorithms for MiDAS (Military Digital Analysis System) using SSE. The interest in the Intel P3 processor stems from the ability to configure them in a dual processor configuration on existing motherboards. Additionally, the use of SIMD instructions provides useful parallelism allowing for significant optimization of signal processing algorithms.

The P3 processor contains 8 additional 128-bit SIMD floating point registers which are directly accessed using XMM registers. The principal use of the 128-bit register is as a packed, single-precision, floating point operand. This allows for parallel storage and computation on 4 single precision floating point numbers. SSE includes 70 functions/extensions to operate on this data type. Packed operations allow for a single operation to manipulate all four floating point numbers at the same time.

According to Intel documents, the Streaming SIMD Extensions set consists of 70 instructions, grouped into the following categories:

- Data movement instructions
- Arithmetic instructions
- Comparison instructions
- Conversion instructions
- Logical instructions
- Additional SIMD integer instructions
- Shuffle instructions
- State management instructions
- Cacheability control instructions

SSE also includes the ability to stream data to memory without polluting cache and to pre-fetch data before it is needed using the cacheability control instructions.

Pentium 3 processors can be divided into two groups according to L2 cache size. The more recent group has 256 KB ATC (Advanced Transfer Cache) which runs at the core speed (full

speed) of the processor. This cache is located on the processor which is commonly referred to as a coppermine P3. Core speed refers to the 'quoted' speed of the processor which is the bus speed multiplied by the clock multiplier. The other L2 cache size available on earlier P3 processors is 512 KB. It runs at half the core speed of the processor and is located off the processor chip.

Table 1: Pentium 3 Processor Cache and Bus Speed Types

P3 Processor	L1 Cache Size (KB)	L2 Cache Size (KB)	L2 Cache Speed (MHz)	Bus Speeds (MHz)
Core Speed Advanced Transfer L2 Cache (ATC) on Chip	32 (16 Data / 16 Infrastructure)	256	1 x Core Speed	100, 133
Half Core Speed L2 Cache off Chip	32 (16 Data / 16 Infrastructure)	512	Core Speed / 2	100

1.3 Summary of Benchmark Timings

Table 2 summarizes the true timings achieved over multiple runs for two P3 processors with the two basic cache types. The processor with the smaller, faster cache always outperforms the processor with the larger and slower cache, but this is mainly due to processor speed. All the timings are impressive compared to historical expectations for FFT speed.

Table 2: Summary of Benchmark Times for Two P3 Systems

Operation Times using the Intel SPL v4.1 on the P3 450 and 733 MHz Processors				
Number of Data Points	Time for 1 FFT, without Data Transfer Effects (µs)		Total Time for 2 Hamming Windows, 2 FFTs and 1 Cross Spectrum, with Data Transfer Effects (µs)	
	P3 450 MHz w/512 KB L2 Cache	P3 733 MHz w/256 KB L2 Cache	P3 450 MHz w/512 KB L2 Cache	P3 733 MHz w/256 KB L2 Cache
32	2	1	6	4
64	3	2	11	7
128	6	3	23	13
256	11	7	46	27
512	23	14	95	53
1024	51	30	179	112
2048	112	66	377	234
4096	314	159	987	530
8192	871	386	3911	1202
16384	2101	924	5558	2864
32768	4784	2463	14168	7842
65536	11728	8139	33073	23022
131072	29260	23573	80372	58200

2 Windowing , FFT Types and Cross Spectrum

2.1 Overview

The algorithms specified by the Scientific Authority for investigation were the complex FFT alone, and a benchmark processing sequence intended to be representative of typical MIDAS detection processing. The benchmark processing sequence consists of the Hamming window, two real FFTs, followed by the Cross Spectrum of the resulting post-FFT data.

The application was written in C++ using MSVC6 (Microsoft Visual C++ 6) as recommended by the Intel SSE development documentation. To take advantage of SSE the Intel C++ compiler was integrated into MSVC using the recommended installation procedure. The Intel SPL (Signal Processing Library) was used for the windowing, FFT and cross spectrum algorithms. As an alternative to using the SPL, an FFT was developed by expanding the Split Radix algorithm from the Intel Architecture Performance Training Library to longer data lengths . This is discussed further in section 2.4.

The windowing, FFT and cross spectrum algorithms were available in the Intel SPL (Signal Processing Library) v4.1 in several formats and are discussed in the following sections. Table 3 summarizes the five major SPL routines used for this work.

Table 3: Intel SPL Functions of Interest

Operation	Intel SPL Function Name	Description
FFT (complex)	nspcfft	Floating Point, Complex, In-Place
Hamming Window	nspswinhamming	Floating Point, Real, In-Place
FFT (real)	nspsrealfft	Floating Point, Real, In-Place
Conjugate	nspcbconj1	Floating Point, Complex, In-Place
Multiply	nspcbmpy2	Floating Point, Complex, In-Place

2.2 Windowing

The first operation typically performed on the data for the MIDAS application is a Hamming window. Both the input and output data arrays are real for the windowing function used. The second row of Table 3 identifies the specific SPL routine used. The window is used to reduce spurious high frequency components for spectral analysis of finite signals.

2.3 SPL FFTs

The complex SPL FFT algorithm on the first row of Table 3 was used for isolated FFT-only benchmarks with a static data array. The real FFT of the third row of Table 3 was used for the typical MiDAS benchmark processing sequence following the windows. The input is real but the output is complex. The same data array is used for both input and output (in-place) so the input is overwritten. The real FFT was selected for the benchmark sequence at the request of the Scientific Authority to model processing of a real IF signal.

2.4 Extended Split Radix FFT

In earlier testing the Intel complex FFT (floating point, in-place) was used for testing and comparison with an FFT developed by the contractor as an extension of the Split Radix algorithm. The complex SPL FFT routine used is listed in Table 3. The SRFFT example is for a complex in-place FFT and was extended to allow data lengths in excess of 128 pts, up to N = 32k pts. It was written in both inline assembly and using C intrinsics and could handle up to N = 32K FFTs. The performance comparison of the SRFFT and SPL is located in section 5.2. The Intel algorithms were used for further testing of the hardware and software effects on speed due to better performance for the data sizes of most interest.

2.5 Cross Spectrum

The cross spectrum operation is performed by taking the conjugate of one FFT result and multiplying it by a second FFT result. The Intel SPL conjugate and multiply operations used are described in Table 3.

The cross spectrum is used to process data from a direction-finding antenna array and the phase values in the cross spectrum can be used to estimate angle of arrival.

3 Software Effects on Speed

3.1 Cache and Pre-Fetching

Intel's SSE technology offers the ability to minimize data access latency by fetching data into cache prior to use so that it is immediately available when needed without main memory data transfer. Ideal scheduling distance (time between pre-fetch call and data usage) is platform and processor dependant due to memory and cache latencies as well as dependence on the amount of data that is pre-fetched. Too long a scheduling distance may cause the data to be pushed out of cache prior to its usage. Too short a distance does not hide the fetch behind computation. Mixing arithmetic instructions and pre-fetches improves performance by avoiding situations which cause instruction buffers to be filled with cache modification instructions. This allows cache modifications to occur during other operations.

Due to the limited scope of the work, pre-fetching was not implemented in these benchmarks. Nonetheless pre-fetching may offer potential for improved performance by allowing main memory data transfer of the next data array to occur in parallel with computation on the current data array.

3.2 Data Alignment in Memory

SSE requires that memory operands be aligned to 16-byte boundaries for optimal performance. Alignment functions are available in the Intel SPL to accomplish this task. This allows four floating point values to map correctly to one XMM register.

3.3 Streaming Stores

Typical operations and movement of data in memory lead to cache pollution with data that will not be accessed in the future. As discussed in section 3.1, having data available in cache at the required time increases performance. Streaming storage allows movement of final output data into memory without placing the data in cache during the process.

Streaming was not implemented in the benchmarks because output data volumes will be small in the MIDAS application. In other applications streaming stores may offer a means to achieve improved performance.

3.4 Order of Operations

As discussed in section 2, the operations performed in the typical MIDAS application are windowing, FFT and cross spectrum. The first operation performed (the window in this case) will eliminate or drastically reduce main memory data transfer for later operations due to caching of data by the first operation. This has been verified and results are shown in section 5.4. Normally operations which access data in a linear fashion are handled by cache loading algorithms better than operations which access data in a more random fashion.

3.5 Assembler, C Intrinsics and Vector Classes

To study FFT performance in isolation, a Split Radix algorithm FFT was developed using code from a 128 point FFT example in the Intel Architecture Performance Training Center 4.5. The original SRFFT was implemented three times over using in-line assembly, C intrinsic functions and vector classes. This example was expanded to perform up to N = 1K FFTs in both in-line assembly and using the C intrinsics. The vector classes were known to be significantly slower, and were therefore neglected. The assembly version performed faster than the C intrinsic version (almost double). Later the assembly version of the SRFFT was expanded to perform up to N = 32K FFTs. The resultant performance exceeded that of the Intel SPL complex, in-place FFT at FFT sizes over N = 4K. Further analysis of performance can be found in section 5.2.

4 Hardware Effects on Speed

4.1 Cache Size

A cache is composed of high speed memory close to the processor and is used to store the most recently used data and instructions.

The smaller L1 (Local One) cache is closest to the processor and provides data with minimum delay. The 32 KB L1 cache is divided into one 16 KB half for data, and one 16 KB half for instructions. When the L1 cache capacity is exceeded, the L2 cache is used. The L2 cache is larger (either 256k or 512K) and is not divided. Once L2 cache capacity is exceeded, main memory is accessed. Relative performance for the all-important FFT should be excellent when the data can fit entirely in either the L1 or L2 cache. When cache capacity is exceeded,

accesses for slower main memory will impede performance.

As the FFT size increases, execution time should increase on the order of N log N until cache capacity is exceeded, beyond which point execution time will grow at a rate higher than N log N.

4.2 Processor Clock Speed

One of the most important factors affecting performance is processor clock speed, which is the internal/core frequency of the processor. The processor clock speed is equal to the bus speed multiplied by the clock multiplier. Although other factors are important, the processor clock speed determines the maximum rate at which calculations can proceed. When data is in cache, the execution time is approximately inversely proportionate to processor clock speed. Clearly a higher processor clock speed is desirable for high speed computation.

4.3 Processor Bus Speed

Processor Bus Speed is the frequency at which the processor communicates with other system components over the system bus. For example, a P3 450 MHz is clocked at a 100 MHz bus speed from the motherboard and its core runs at that frequency multiplied by the clock multiplier (in this case 4.5). Therefore the processor runs at 450 MHz internally and 100 MHz externally. In the case of a P3 processor with 256 KB of L2 cache, the cache runs at the same frequency as the processor, whereas on the 512 KB 'half-speed' L2 cache processors, the cache runs at half the core frequency, or 225 MHZ.

The difference in bus speed between 100MHz and 133MHz processors has an effect on the number of processor clocks taken to complete an operation when the data does not fit into cache. For large dynamic data sets (which exceed cache and must be transferred from main memory over the processor bus), the faster bus speed will decrease data transfer time. Therefore a faster processor will often benefit from increased bus speed so that the effect of data transfer is reduced once cache limits are surpassed.

4.4 Data Transfer Issues

The order of operations in an application influences data transfer rates. Data transfer delay is eliminated or drastically reduced for later operations due to caching of data by the first operation.

Experiments show that data transfer time during windowing is less than the transfer time during FFT. This is likely due to better cache performance for the linear access in the windowing operation compared to non-linear bit reversed memory access in the FFT. As discussed in section 3.1, cache prefetching may offer the possibility of hiding data transfer behind processing so that the data transfer delay is avoided by having it occur in parallel with processing.

Exceeding cache limits makes the application more dependant on data transfer as a performance limitation. Once this occurs a faster processor spends more clock ticks waiting

for data than a slower one. The realized speed advantage of the faster processor is then reduced. To prevent this a faster bus speed is desirable.

5 Testing and Results

The purpose of this section is to describe experiments conducted to determine the impacts of critical factors described in sections 3 and 4, and the relative times for data transfer, on the various operations in a typical processing sequence. Overall timings in microseconds have already been presented in Table 1 using two typical processor configurations for the complex FFT and for a benchmark processing sequence consisting of windowing, FFTs, and cross spectrum calculations.

5.1 Overview

The tests presented in section 5 fall into two categories:

1. FFT timings as a function of FFT length, N
2. Timing breakdowns for the benchmark sequence as a function of data length, N

The FFTs of interest were the two SPL routines of section 2.3 and the extended split radix algorithm of section 2.4. The benchmark sequence consisted of two hamming windows, two FFTs and a cross spectrum as used by the MiDAS system.

Four P3 processors (450, 500, 700 and 733 MHz) were tested. The 450 and 500 had 512 KB of half core speed L2 cache and the 700E and 733EB had 256 KB of core speed L2 cache. Some of the tests were run with L2 cache disabled to determine cache effects upon results.

The 450, 500 and 700E had 100 MHz buses and the 733EB had a 133 MHz bus. The 733 was used to determine the effect of faster bus speeds upon results.

Results were obtained by timing specific sections of the application during operation. Three timing types (illustrated in Figure 1) were required to extract the timing information.

Timing type A was used to time individual signal processing operations in isolation. Type A timings were made for applications operating on a static data set for each loop (which would remain in cache) and for the same applications operating on a portion of a large data set that was loaded at the start of each loop (the data would have to be transferred into cache). The timings were compared to determine the effect of data transfer for individual functions and to verify the timings of individual operations in the benchmark processing sequence.

Timing type B was used to obtain the total time for a benchmark sequence of operations consisting of windowing, FFT, and cross spectrum. Timing type B was also run using both static and dynamic data sets (i.e. new input data loaded at the start of each loop). The timings with dynamic data sets include cache data transfer overhead.

Timing type C was used to time individual operations within the benchmark sequence for both static and dynamic input data sets. We found that when using a dynamic data set which must be transferred to cache, operation 1 will suffer from data transfer but data transfer will be eliminated or drastically reduced for the other operations in the benchmark sequence.

To get representative numbers of clock ticks each experiment was run in four series of up to 5000 runs each for data arrays from N = 32 point to 128K points. For data lengths above 4K, the number of runs per series was reduced to avoid hard disk transfers when memory was exceeded. The number of runs per series and a complete table of the tests performed can be found in Appendix A: Summary Table of Tests.

In Figures 2, 3 and 4 each of the four series is plotted separately as a different symbol (all in the same colour) representing the modal value for the execution time over all the runs in that series. The series are plotted separately to verify consistency and to ensure that the mode was not affected by random system interrupts which lead to spurious readings for execution time.

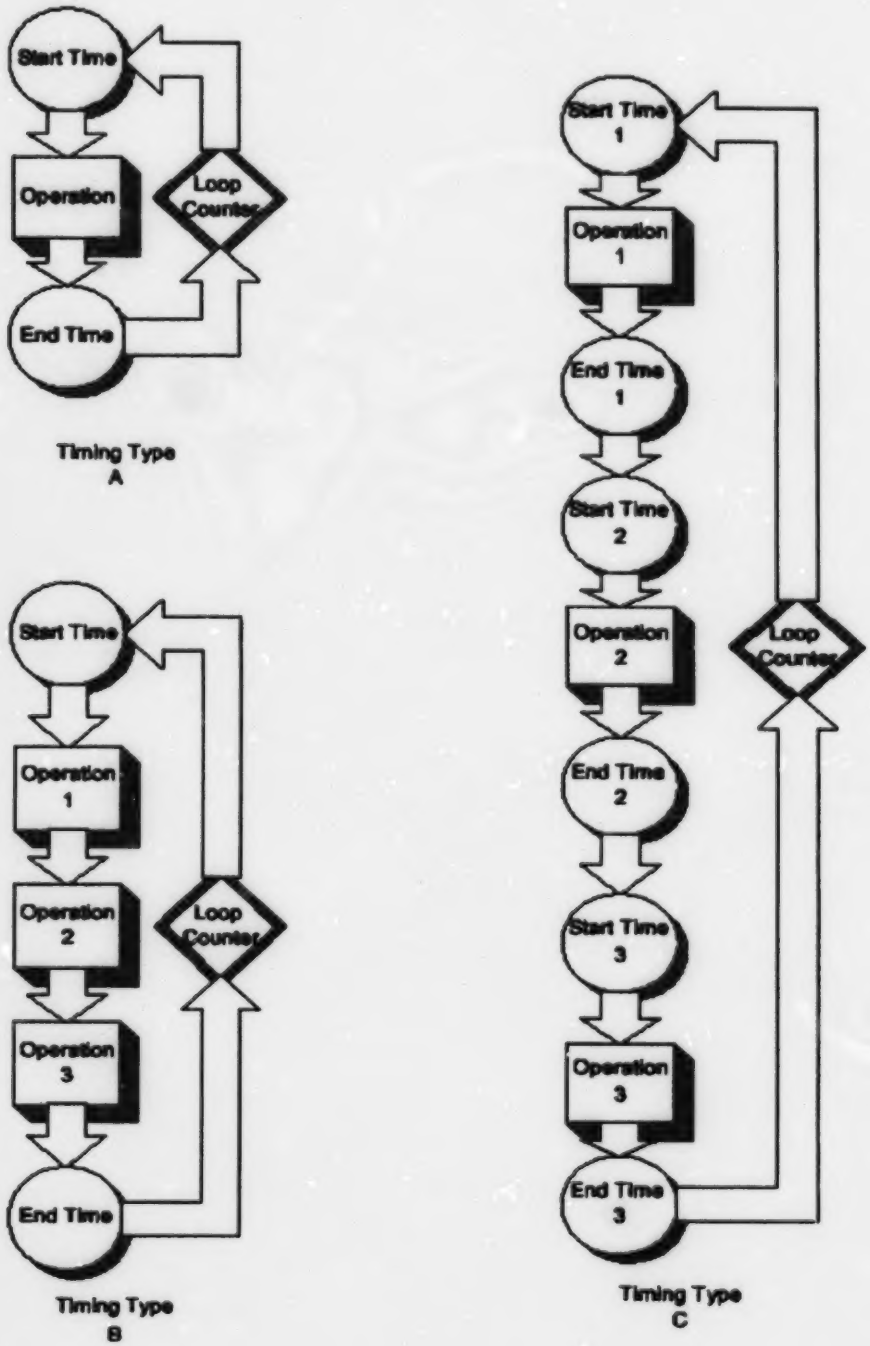

Figure 1: Benchmark Timing Types

5.2 Expanded SRFFT vs Intel SPL Complex FFT

Comparison tests showed that the Intel SPL 4.1 complex FFT performed faster than the SRFFT at data sizes less than N = 4K (log2(4096) = 12). However the complex SRFFT was faster than the SPL version above N = 4K. Preliminary attempts to improve performance in the SRFFT using pre-fetching were unsuccessful and lead to diminished performance.

By normalizing the number of clock ticks by NlogN, we are able to compare relative efficiency in processing complex FFTs for different values of N.

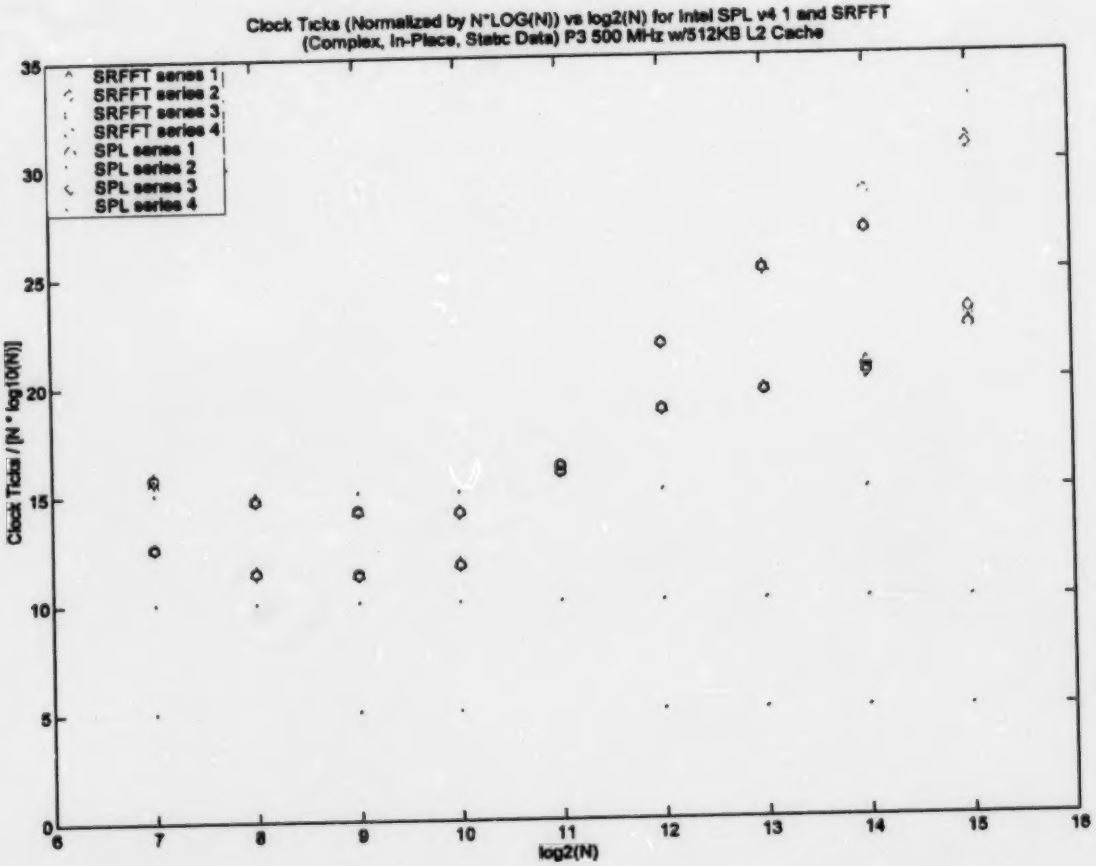

Figure 2: Comparison of Expanded Split Radix and Intel Complex FFTs

5.3 Cache and Clock Speed Effects for Intel Real FFT

In order to determine the effect of cache overflow, the length of the real input array was increased by factors of two and the execution time in clock ticks was measured and plotted in normalized form.

In Figure 3 the L2 cache was disabled for all processors, so that only the L1 cache is active. The normalized real FFT execution time in clock ticks is plotted versus the log of data length (i.e. FFT size). The four processors of interest are plotted in four different colours and each of the four series results appears as a separate symbol.

Note that below 4K points (log2(4K) = 12), all the processors closely track each other. This is because all the data fits easily into cache, and because the measurement of time in clock ticks eliminates the effect of clock speed. The smallest normalized value occurs at the 512 point size, indicating that this is where the FFT is most efficiently computed for L1 cache only.

At 4K points, normalized clock ticks increases sharply, indicating that the L1 cache is overflowing (i.e. the cache limit is being exceeded routinely, forcing more main memory accesses). This makes sense because at 4K points and 4 bytes per floating point data point, the input array takes up the entire 16 KB data portion of the L1 cache.

After the cache limit is exceeded (i.e. at 4K points and above) the processor curves start to diverge. This is due to the fact that memory accesses are occurring and that a different number of clock ticks will occur during memory accesses for processors of different speeds. A faster processor will have more clock ticks during a memory access than a slower processor. Note that the 733 MHz processor differs from the other three in that it uses a 133 MHz memory bus rate, while the other three processors all use 100 MHz bus rate.

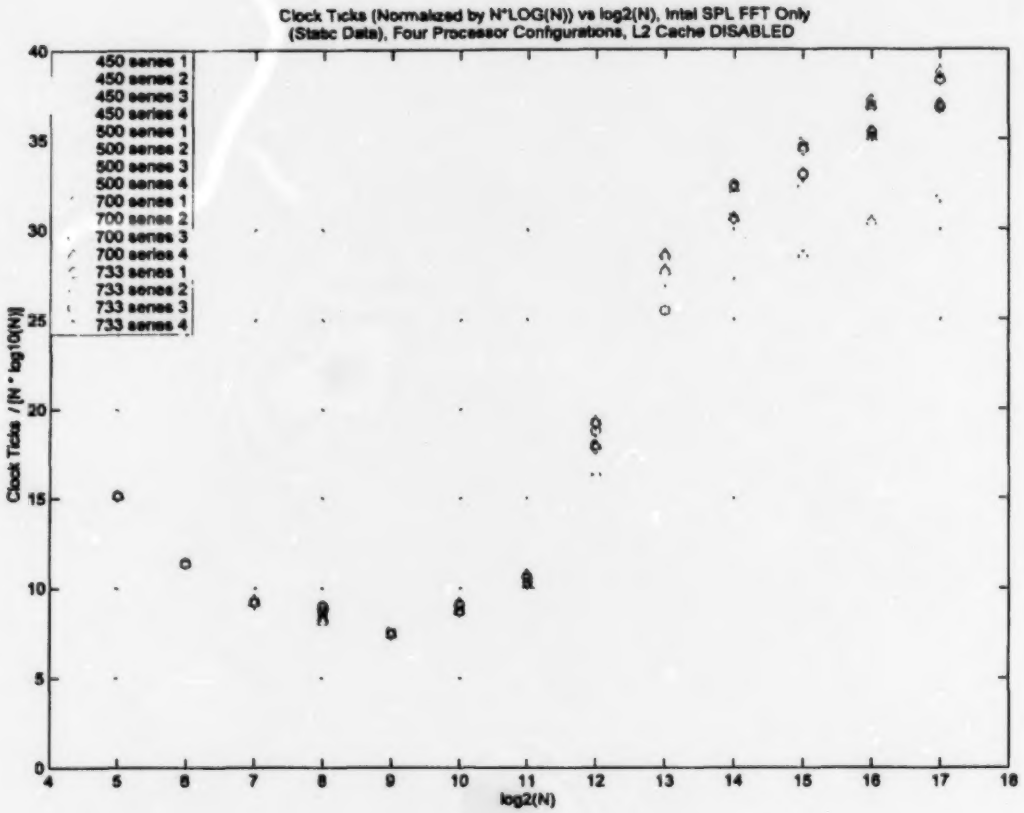

Figure 3: Intel Real FFT Performance for Four Processors, L2 Cache Disabled

In Figure 4 the experiment is re-run but with the L2 caches enabled. Note that of the four processors, the two with lower processor clock rates use the half core speed 512 KB cache, and the two with higher processor clock rates use the full core speed 256 KB L2 cache.

There is still an increase in normalized clock ticks above 4K data points, but the rise is more gradual than with the L2 caches disabled. The 700 MHz and 733MHz processors rise very gradually above the 4K point because they have a faster L2 cache. The 450MHz and 500MHz processors rise significantly faster until a crossover point is reached at 64K points.

At N=64K (log2(64K) = 16) and above, the 700 and 733 MHz processors suddenly increase sharply in normalized clock ticks. This is because their 256 KB L2 cache capacity has been exceeded. At the 64K and 128K point FFTs, the processors with the larger cache are actually more efficient in terms of clock ticks and would actually run faster if the processor clock speeds were matched.

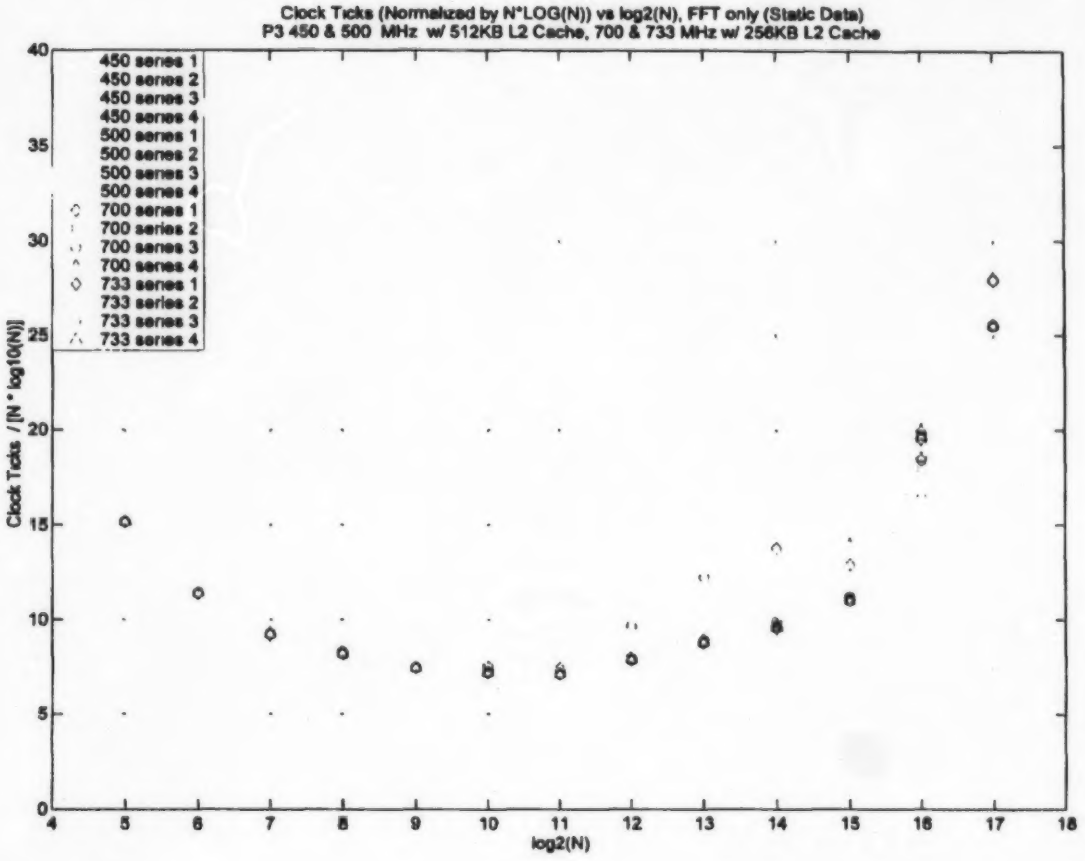

Figure 4: Intel Real FFT Performance for Four Processors, L2 Cache Enabled

5.4 Performance of Typical MIDAS Processing Sequence

Figure 5 is a stacked bar chart showing the relative time taken for each of the four major categories of functions that make up the benchmark sequence. Time is presented in clock ticks normalized by N log N.

The valley of this bar chart occurs at the 2K, 4K and 8K data array lengths, indicating that these are the most efficiently processed. Data transfer during the initial windowing operation is defined as the difference between the window execution ticks for dynamically changing data and the window ticks for static data. This data transfer is a significant portion of total clock ticks up to about 16K points. Above 32K points all components rise quickly due to cache overflow except for window data transfer. At the large values the data transfer number begins to lose meaning since even the static data set cannot fit in cache at 64K points.

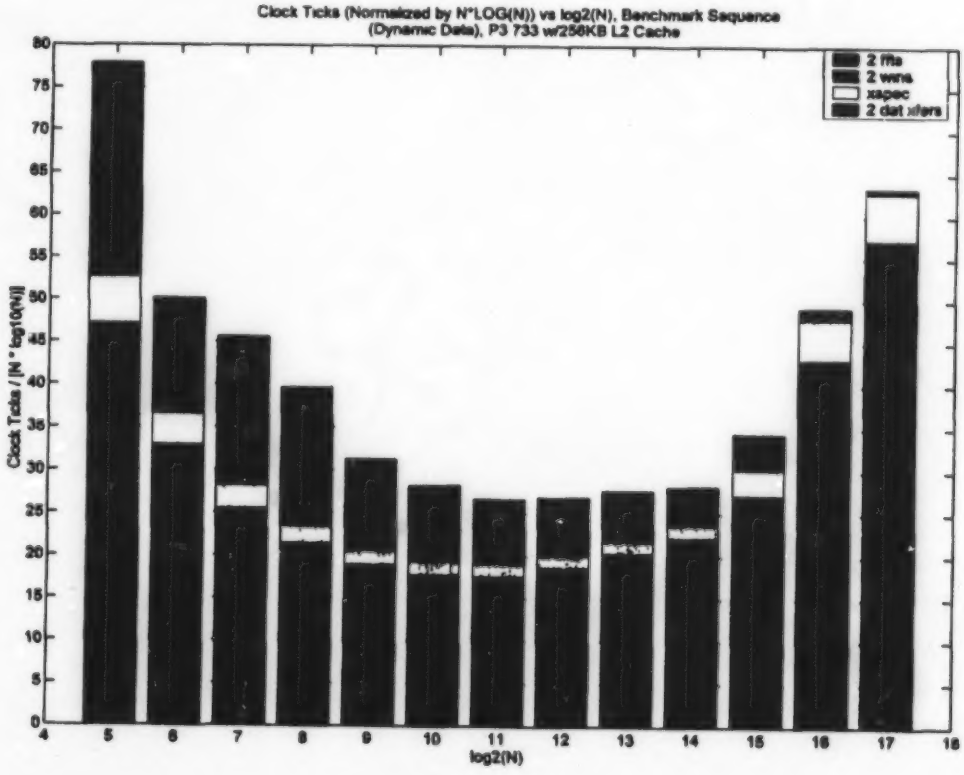

Figure 5: Performance of Components of the Benchmark Processing Sequence

Figure 6 shows the same data as the Figure 5, but normalizes the bar height to 100% so that the relative importance of each component can be seen. Note that the FFT increases in importance with increasing array size because it grows on the order of NlogN, while the other operations grow linearly.

Also note that initial data transfer becomes less important at the higher data lengths, largely because many cache misses are occurring during the program execution. The defined data transfer in these cases is probably being incorporated into the other operations since cache misses will start to occur during and between operations.

For the 1K and 2K FFTs of primary interest, the FFT execution takes on the order of one half of the total execution time.

Of the other 3 components in figure 6, the initial data transfer is the largest. Therefore speeding up initial data transfer would be beneficial in order to reduce overall execution time.

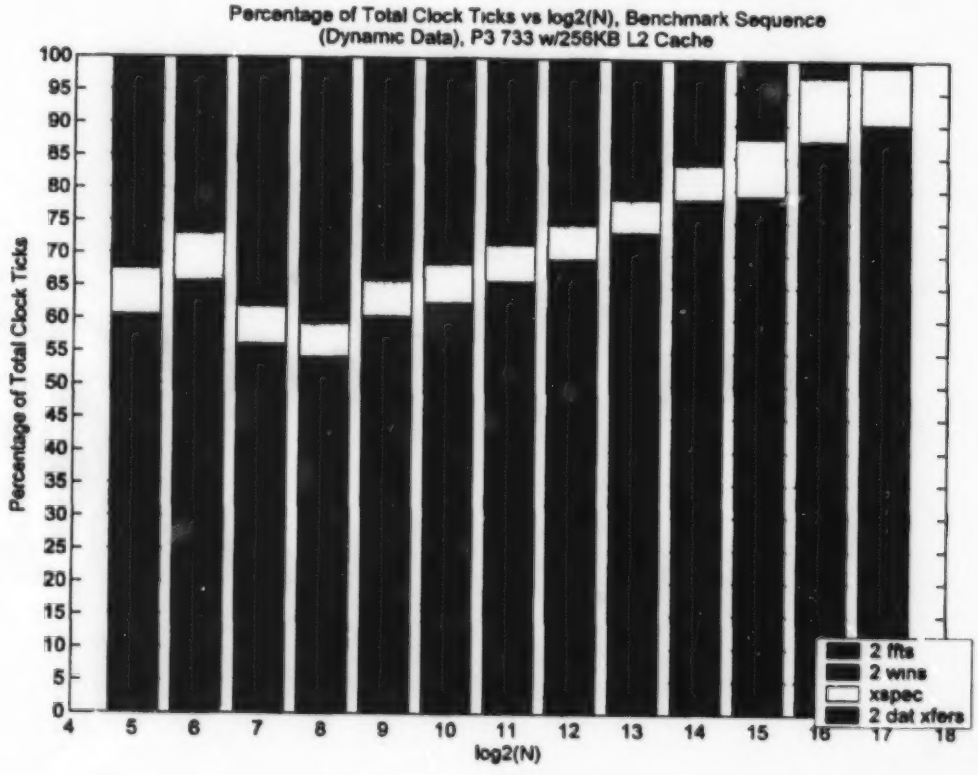

Figure 6: Relative Importance of Components of the Benchmark Sequence

6 Conclusions

The following conclusions can be drawn from the work under this project:

1. The Intel SPL floating point complex in-place FFT is faster than the SRFFT below N = 4K but slower above N = 4K.

2. In-line assembly code outperforms corresponding C intrinsics for the SRFFT by approximately a factor of two.

3. The Intel SPL contains a useful library of routines that allow the developer to take advantage of SSE with a minimum of effort.

4. Cache size relative to data length is a key determinant of performance. FFT performance is decreased due to a L1 cache limit at the N = 4K FFT. The L2 cache limit occurs at N = 32K on a 256 KB L2 cache processor. A noticeable increase in time occurs at the N = 64K mark on a 512 KB L2 cache processor, which is likely the beginning of the L2 cache limit.

5. The newer 256 KB full core speed cache outperforms the older 512 KB half speed cache for the data lengths of interest. This form of the Pentium 3 processor is further preferred because the higher processor clock rates are available only with the full speed cache.

6. The Intel SSE architecture performs the benchmark processing sequence most efficiently at the 2K, 4K and 8K data lengths, but also performs very well from about 512 points up to 16K points.

7. For the moderate size data records of interest, overhead for data transfer to cache is incurred only on the first operation, such as the window operation in the benchmark processing sequence.

8. After the FFT itself, initial data transfer time is the most important bottleneck, and remains significant (about 30%) at the 1K and 2K data lengths of most interest.

9. Surpassing cache limits makes the application data transfer dependant. Once this occurs a faster processor spends more clock ticks waiting for data than a slower one, reducing the efficiency of the faster processor. A faster bus speed would increase data transfer rates during processing.

10. SIMD facilitates impressive performance for signal processing applications and the P3 processor architecture is well-suited to the signal processing operations of interest.

7 Recommendations

The 256 KB full-speed cache processors are recommended for the MiDAS application since L2 cache overflow does not occur until N = 32K, while the data lengths of interest are 1K and 2K points. Current P3 processors all use the smaller, faster 256 KB cache, so this choice also means that faster processor clock speeds will be available.

The faster the processor is, the faster it can execute instructions until L2 cache is exceeded. When this occurs, data transfer becomes an issue of concern and reduces efficiency. More importantly, data transfer of the initial data is a significant factor even when cache overflow does not occur. A faster bus speed will increase data transfer rates and boost performance. Cache pre-fetching of the next array of input data should also be investigated to reduce data transfer overhead.

Computation of power and phase from the cross spectrum results were also operations of interest which were not pursued in favour of extension of the Split Radix FFT. Power and phase information can be computed using the Intel SPL functions in Table 4: Intel SPL Power and Phase Functions.

Investigation into AMD's Athlon processor and its 3DNow! instructions is recommended if the P3 performance achieved in this study is perceived to be inadequate or marginal. Some Athlon processors have 128 KB L1 cache compared to P3s 32 KB, which should be beneficial. Industry reports indicate that the Athlon performs well against the P3, though the depth of software support remains to be determined.

The Intel SPL is a relatively easy-to-use means of getting the P3s SSE benefits without specialized SSE software development. The Intel real FFT provided performs well compared to the SRFFT for the 1K and 2K FFTs of most interest. If further performance improvements are desired using the P3, new custom software development could also be undertaken:

1. An SSE FFT implementation which efficiently combines a Hamming Window with the FFT operation to eliminate the explicit windowing operation.
2. An SSE implementation of the FFTW ("Fastest FFT in The West") algorithm, which was used in MATLAB 6 with excellent results. This may yield improvements for the single most time consuming operation in the benchmark sequence.
3. An SSE implementation of a fast four-fold parallel FFT throughput using SSE. This would make full use of the 128-bit data types (four single precision floats) for all calculations and may allow near peak performance for more of the time.
4. Use of so-called data folding where the data array length is halved in a special way just after windowing. The folding consists of breaking the array into an upper and lower half, and then adding the two halves point-by-point. This allows a half-length FFT to be used and while preserving signal to noise ratio at slightly reduced frequency resolution.
5. Addition of data pre-fetching instructions to initiate cache loading in parallel with FFT computation so that the next data array is ready in cache when the current data has been processed.

6. The Hamming Window could be custom implemented in SSE since the current SPL implementation takes more time than a cross spectrum operation. This would seem to indicate that Intel implementation is not optimized for speed.

Table 4: Intel SPL Power and Phase Functions

Intel SPL Function Name	Description
nspcbphase	complex input (real and imag alternating) in first array, real phase output in second array
nspsbrphase	real and imag input in first and second array, real phase output in third array
nspcbmag	complex input (real and imag alternating) in first array, real magnitude output in second array
nspsbrmag	real and imag input in first and second array, real magnitude output in third array

ACRONYMS, ABBREVIATIONS, DEFINITIONS

ATC	Advanced Transfer Cache
FFT	Fast Fourier Transform
In-Place	Replacement of input data with output data in variables
MiDAS	Military Digital Analysis System
MMX	Intels 64-bit Data Registers, Instructions or Technology
SIMD	Single Instruction Multiple Data
SPL	Signal Processing Library
SRFFT	Split Radix FFT
SSE	Streaming SIMD Extensions
XMM	Intels 128-bit Floating Point Registers, Instructions or Technology

REFERENCES

[1] Intel Architecture Software Developer's Manual, Volume 2, pp. 9-9 - 9-19

[2] Intel Architecture Optimization Reference Manual, Ch 6

[3] Intel Architecture Optimization Reference Manual, Appendix D

[4] Intel C/C++ Complier User's Guide: With Support for the SIMD Extensions, Ch 13

[5] AP-589 Software Convention for the Streaming SIMD Extensions, V2.1, Jan, 1999

[6] AP-900 Identifying Support for the Streaming SIMD Extensions in the Processor and Operating System, V1.1, Jan, 1999

[7] Intel Signal Processing Library Reference Manual, Document # 630508-012

P515881.PDF [Page: 32 of 37]

APPENDIX A: Test Summary Table

Operation Tested	# Data Points	Data Type	# of Series	# of Runs per Series	P3 Processor Speed / L2 Cache	L2 Cache Enabled
SRFFT	128-32K	Static Floating Point Array (of size data points)	4	5000	500 MHz / 512 KB	yes
						no
					733 MHz / 256 KB	yes
						-
Intel SPL complex in-place FFT	128-64K	Static Floating Point Array (of size data points)	4	5000	500 MHz / 512 KB	yes
						no
Intel SPL real, in-place hamming window	32-128K	Static Floating Point Array (of size data points)	4	5000	450 MHz / 512 KB	yes
						no
					500 MHz / 512 KB	yes
						no
					700 MHz / 256 KB	yes
						no
					733 MHz / 256 KB	yes
						no
	32-128K	Dynamic Floating Point Array (of size data points)	4	up to 4K=5000, 8K=2400, 16K=1200, 32K=600, 64K=300, 128K=50	450 MHz / 512 KB	yes
						-
					733 MHz / 256 KB	yes
						-

Intel SPL real, in-place FFT (Both alone and post Hamming Window)	32-128K	Static Floating Point Array (of size data points)	4	5000	450 MHz / 512 KB	yes
						no
					500 MHz / 512 KB	yes
						no
					700 MHz / 256 KB	yes
						no
					733 MHz / 256 KB	yes
						no
	32-128K	Dynamic Floating Point Array (of size data points)	4	up to 4K=5000, 8K=2400, 16K=1200, 32K=600, 64K=300, 128K=50	450 MHz / 512 KB	yes
						-
					733 MHz / 256 KB	yes
						-
Intel SPL complex multiply and conjugate (cross spectrum) (Both alone and post FFT)	32-128K	Static Floating Point Array (of size data points)	4	5000	450 MHz / 512 KB	yes
						-
					733 MHz / 256 KB	yes
						-
	32-128K	Dynamic Floating Point Array (of size data points)	4	up to 4K=5000, 8K=2400, 16K=1200, 32K=600, 64K=300, 128K=50	450 MHz / 512 KB	yes
						-
					733 MHz / 256 KB	yes
						-

APPENDIX B: Platform Specifications

Platform 1 (for 450, 500 and 700 MHz processors):

Motherboard:	Aopen AX6BC
Chipset:	Intel 440 BX AGPset
System Memory:	192 MB
BIOS:	Award revision 2.53, 03/29/2000

Platform 2 (for 733 MHz processor):

Motherboard:	ASUS CUV4X
Chipset:	VIA Apollo Pro133Z
System Memory:	256 MB
BIOS:	

BIOS Chipset Settings	Platform 1	Platform 2
SDRAM CAS Latency	3T	2T
SDRAM RAS# to CAS# Delay	2T	2T
SDRAM RAS# Precharge	2T	2T
SDRAM Precharge	Disabled	5T
DRAM ECC	Disabled	NA
System BIOS Cachable	Disabled	NA
Video BIOS Cachable	Disabled	NA
Video RAM Cachable	Disabled	Disabled
8 bit I/O Recovery Time	4	NA
16 bit I/O Recovery Time	1	NA
Memory Hole	Disabled	NA
Passive Release	Disabled	NA
Delayed Transaction	Disabled	Disabled
AGP Aperature Size	64 MB	64 MB
PII Microcodes	Enabled	NA
Clock Spread Spectrum	Off	NA
Bus Speed	100 MHz	133 MHz
Read Around Write	NA	Enabled

CPU DRAM Back-Back Transfer	NA	Enabled
PCI to DRAM Prefetch	NA	Disabled
Byte Merge	NA	Disabled
PCI/VGA Palette Snoop	NA	Disabled
PCI Latency Timer	NA	32
L2 Cache ECC	NA	Enabled
AGP Capability	NA	4X
DRAM Read Latch Delay	NA	1.5 ns
Bus Speed	100	133
Clock Multiplier	4.5	5.5

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1 ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
NATIONAL DEFENCE
SHIRLEYS BAY, OTTAWA, ONTARIO K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

ANALYSIS OF SIMD SIGNAL PROCESSING PERFORMANCE (U)

4 AUTHORS (Last name, first name, middle initial)

Brinson, Bill and Saper, Ron

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
March 2001	24	7

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contractor Report

8 SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
NATIONAL DEFENCE
SHIRLEYS BAY, OTTAWA, ONTARIO K1A 0Z4 CANADA

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)
5bb12	W7714-000350

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNLIMITED

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

DCD03 2/08/87

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual)

The Single Instruction Multiple Data (SIMD) instruction set extensions introduced on the PIII processor are of potential importance in signal processing and other application. This report presents the results of an initial investigation for two implementations of the Fast Fourier Transform (FFT) into the achievable performance benefits and their dependence on factors such as cache memory size and speed.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Signal processing
Fast Fourier transform (FFT)
SIMD
SSE

#515881

CA011295

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM